PlatformConference

Direction • Design • Perspective • Analysis

HyperTransport Technology : A Tutorial

Jay Trodden

MindShare, Inc.

www.mindshare.com

jay@mindhsare.com



January 23-24, 2002

Topics

- HyperTransport Scope
- IO Interconnects: Background
- HyperTransport Features
 - Device Types And Topologies
 - Basic Link Features
 - Scalability: Link Width And Clock Speed
 - Protocol Basics: Packet Structure
 - Protocol Basics: Requests
 - Protocol Basics: Response/Data
 - Managing The Links: PCI Configuration
 - Managing The Links: Flow Control
 - Managing The Links: Error Handling

Copyright MindShare Inc., 2002

HyperTransport Scope

- Intended as a high-performance “inside the box” motherboard IC component connection. It addresses a number of limitations in shared bus connections.
- Targets IT, Telecom, and other applications requiring high bandwidth, scalability, and low latency.
- Formerly known as AMD’s Lightning Data Transport (LDT), HyperTransport is now backed by a consortium of developers.

Copyright MindShare Inc., 2002

PlatformConference

Direction • Design • Perspective • Analysis

IO Interconnects: Background



January 23-24, 2002

Demands On IO Subsystems

- A number of factors strain the ability of IO subsystems to keep pace with applications:
 - Ever-increasing CPU internal and external bus clock speeds
 - System topologies (including multiple CPUs) running applications which emphasize high bandwidth IO access over processing.
 - Faster peripheral devices which see bottlenecks when using the IO subsystem to reach system memory in support of their applications.

Copyright MindShare Inc., 2002

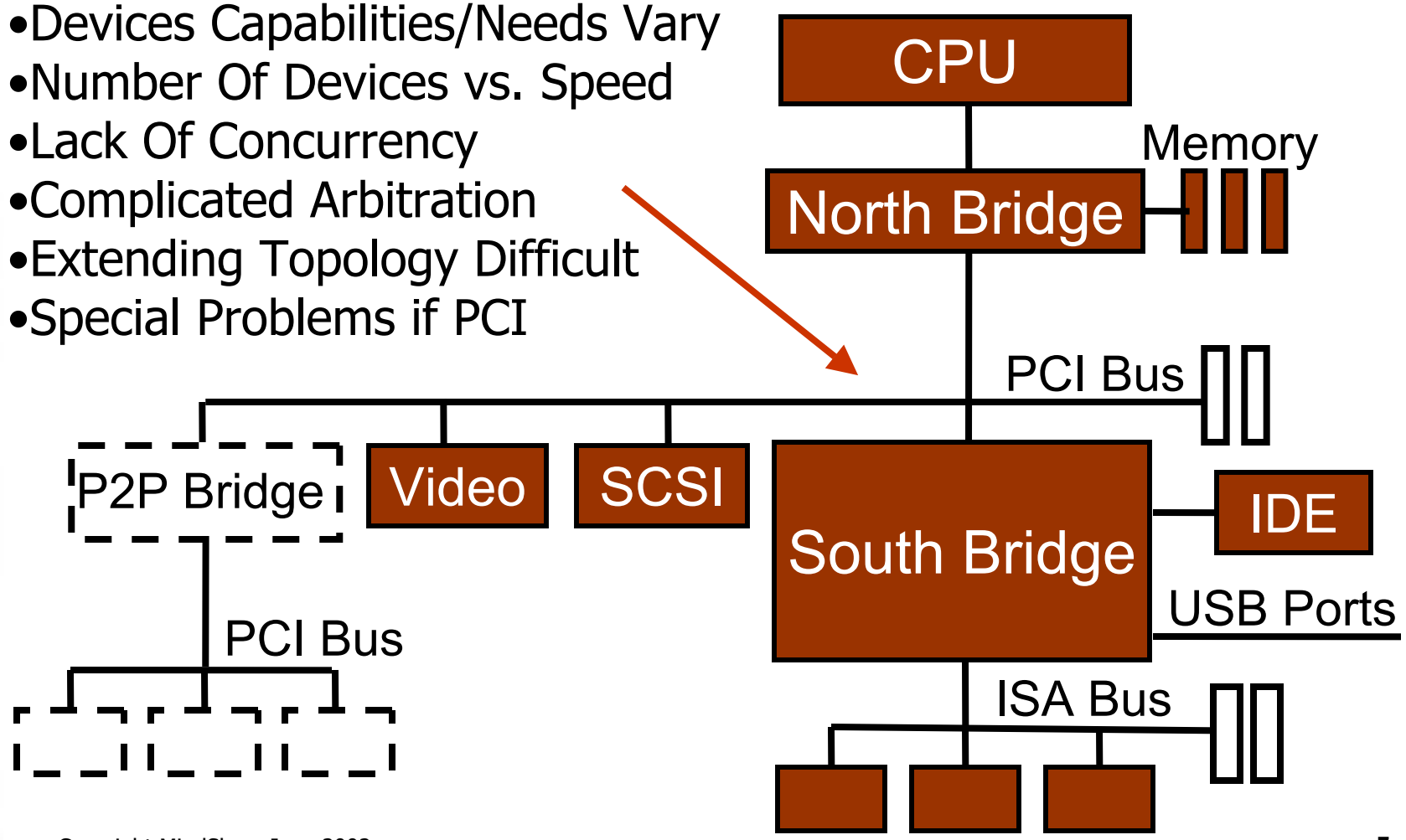
Shared Bus vs. Point-Point IO

- Two important schemes are used to connect IO subsystems to memory:
 - Shared bus connections, such as PCI, PCI-X
 - Point-point component connections, which include HyperTransport and some proprietary hubs.

Copyright MindShare Inc., 2002

Limitations Of A Shared Bus

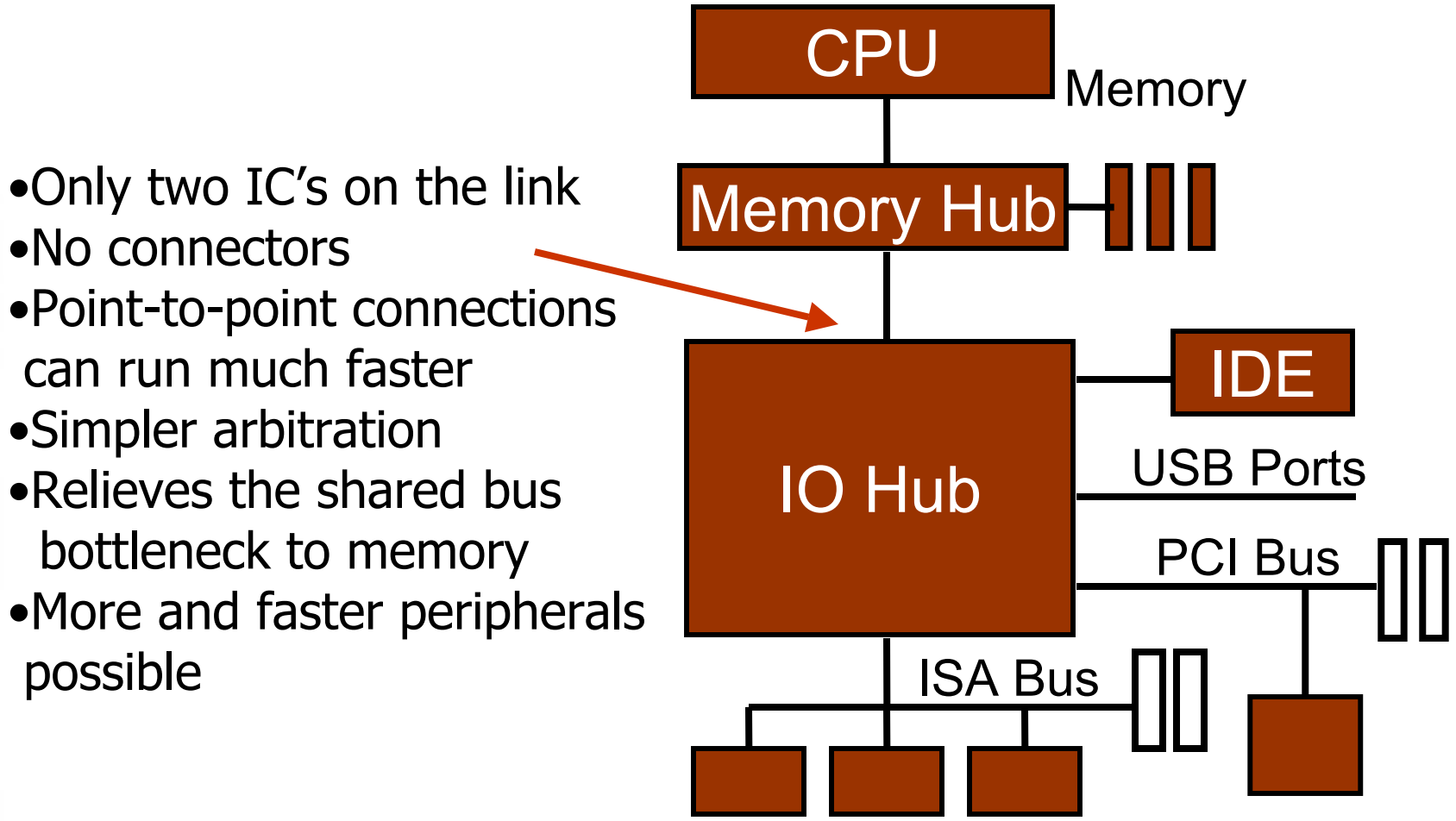
- Many systems use some variation of a *North-South* bridge IO interconnection:



Copyright MindShare Inc., 2002

Component Connections Help

- Some newer systems replace shared bus with component connections on critical links:



- Only two IC's on the link
- No connectors
- Point-to-point connections can run much faster
- Simpler arbitration
- Relieves the shared bus bottleneck to memory
- More and faster peripherals possible

Copyright MindShare Inc., 2002

Platform Conference

Direction • Design • Perspective • Analysis

HyperTransport Features



January 23-24, 2002

What HyperTransport Brings

- A fast, non-proprietary component connection.
- *Tunnel* and *bridge* devices allow creating a HyperTransport system fabric comprised of independent, customized links.
- Scalability. Link width (translates to pin count) and clock speed may be scaled to suit device performance/cost goals.
- A packet-based protocol that eliminates transaction retries and disconnects.
- Minimal impact on OS and driver software
- Compatibility with legacy PC's: configuration, interrupts, power management, etc.

Copyright MindShare Inc., 2002

Platform Conference

Direction • Design • Perspective • Analysis

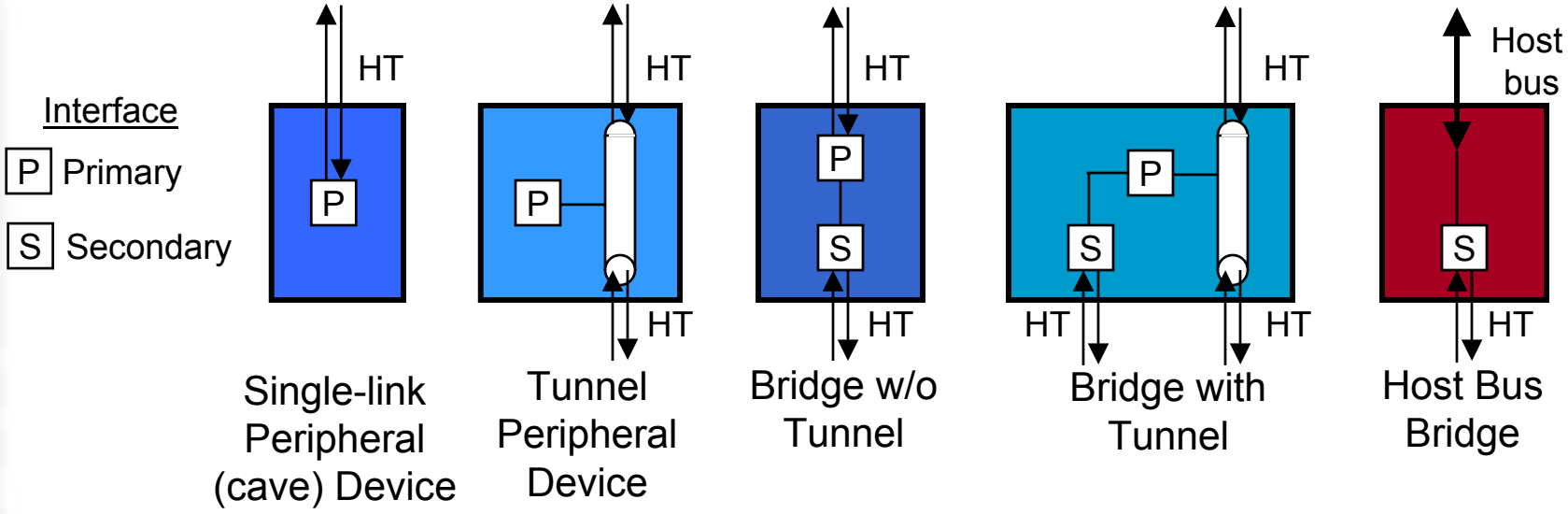
Device Types And Topologies



January 23-24, 2002

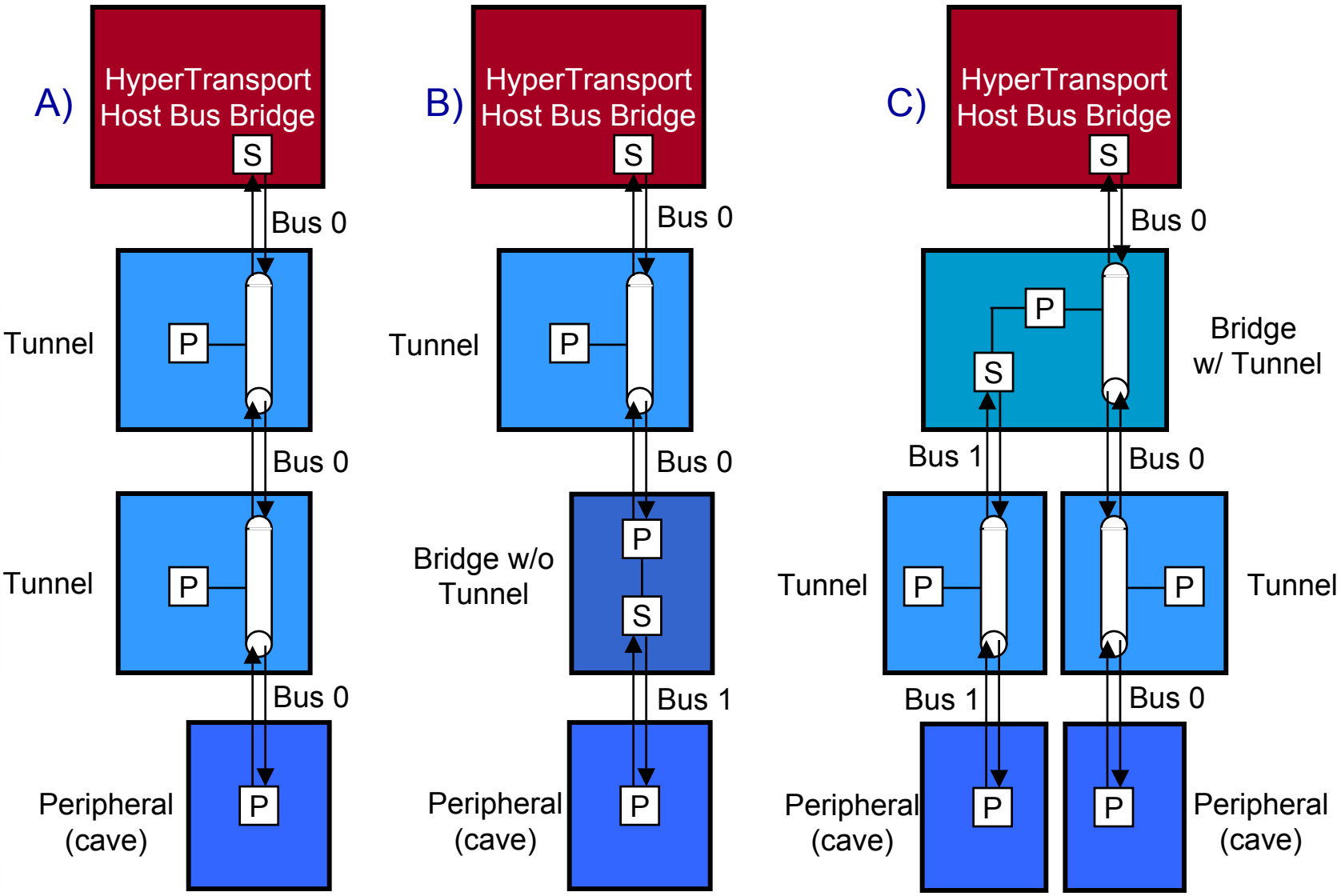
HyperTransport Device Types

- Peripheral (cave) device with a single link
- Tunnel peripheral device
- Bridge device w/o tunnel
- Bridge device with tunnel
- Host bus bridge



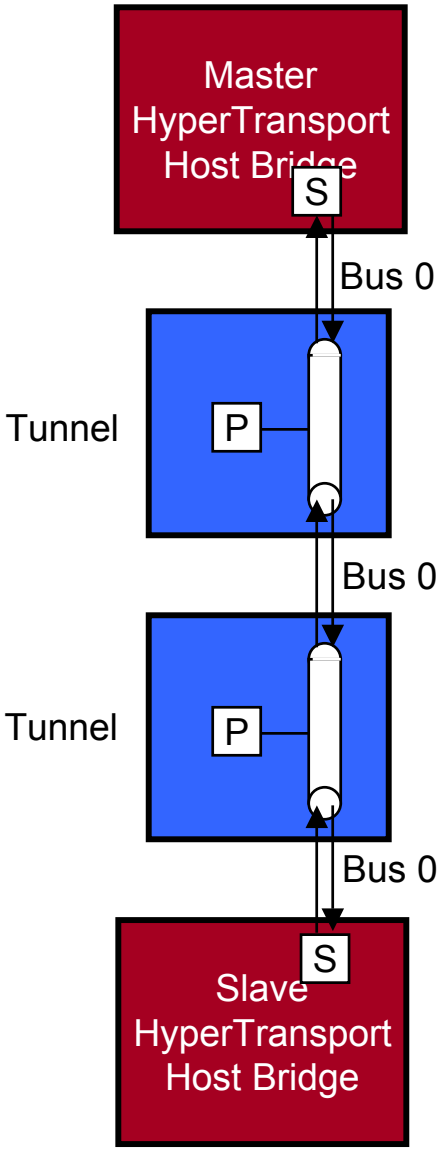
Copyright MindShare Inc., 2002

Some HyperTransport Topologies



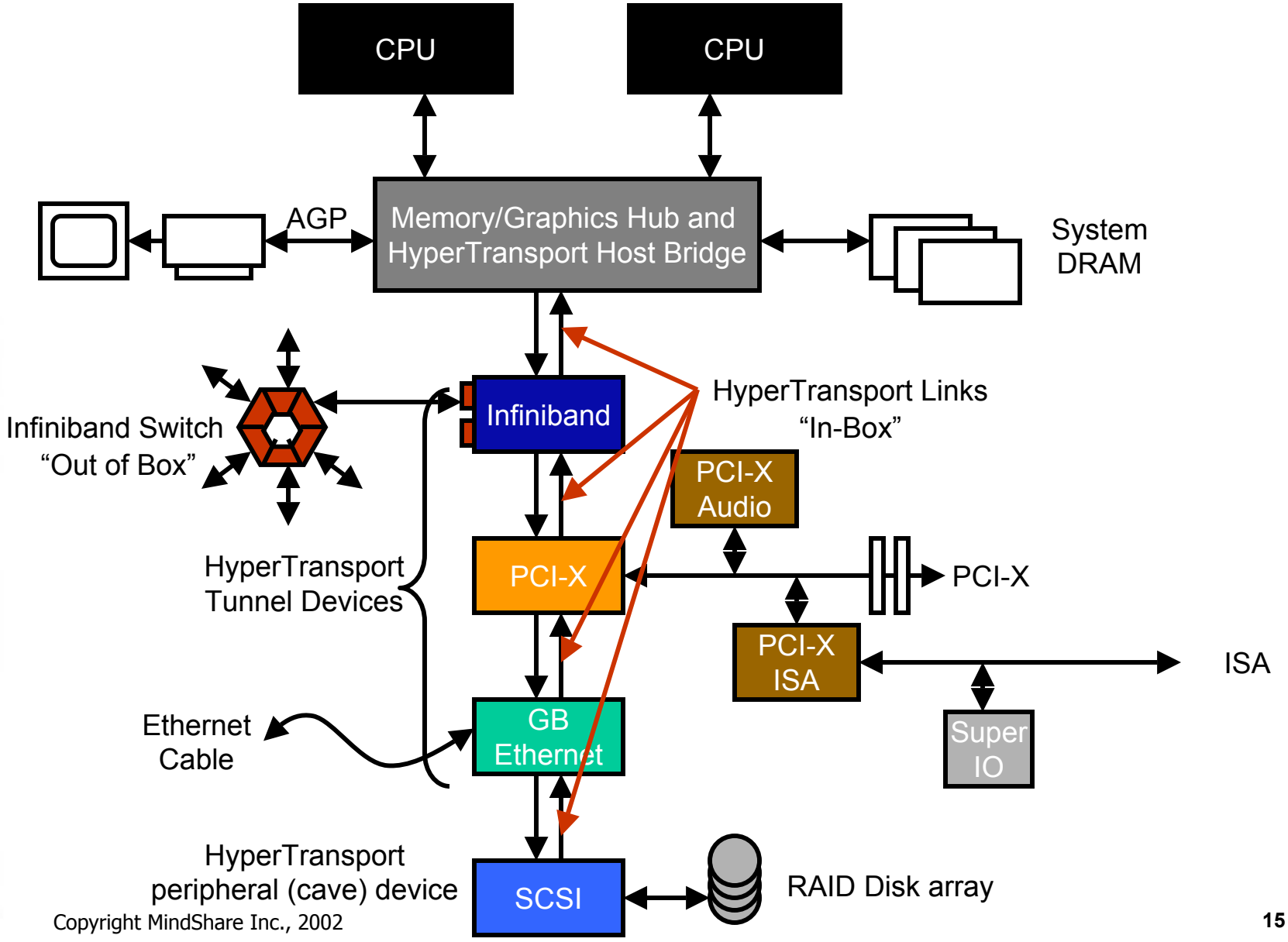
Copyright MindShare Inc., 2002

Double-Hosted Chain Topology



Copyright MindShare Inc., 2002

Example HyperTransport System



Copyright MindShare Inc., 2002

PlatformConference

Direction • Design • Perspective • Analysis

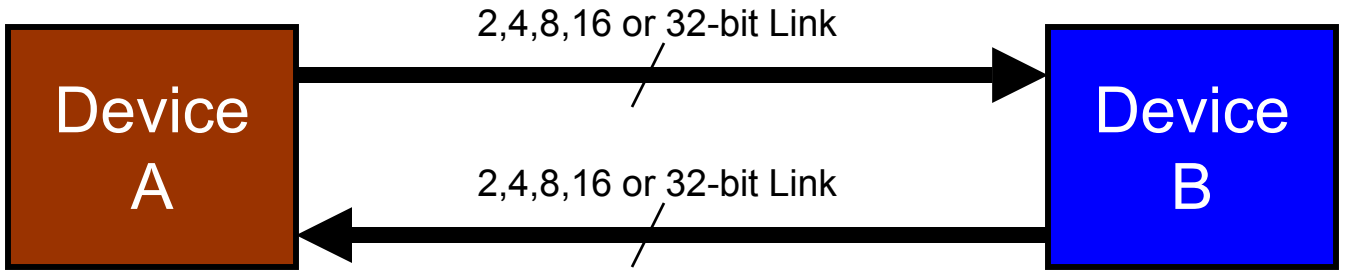
Basic Link Features



January 23-24, 2002

The HyperTransport Link

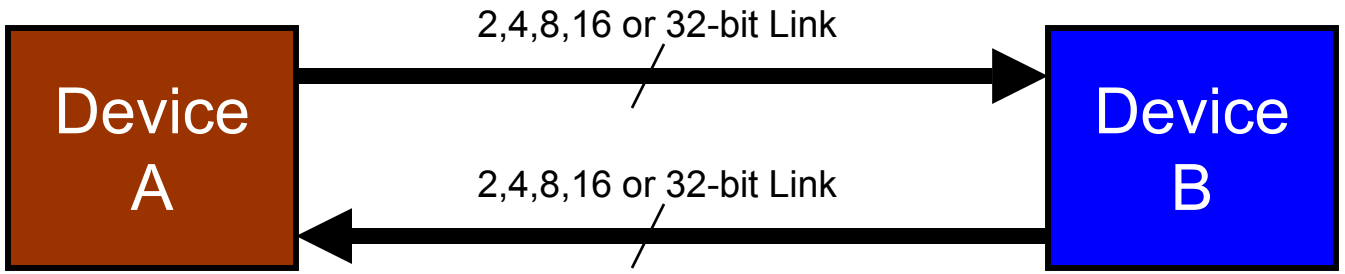
- HyperTransport is a packet-based I/O link between two devices (point-to-point), based on two unidirectional sets of signals.
- Link width can be 2, 4, 8, 16, or 32 bits wide in each direction, and asymmetrical widths are allowed



Copyright MindShare Inc., 2002

Key Signal Groups On The Link

- **CAD** (command, address, data). Multiplexed signals carry request, response, information, and data packets.
- **CTL** (control). Used to indicate data vs. control packets moving across the CAD signals.
- **CLK** (clock). Synchronous HyperTransport clock used in moving CAD and Control.



Copyright MindShare Inc., 2002

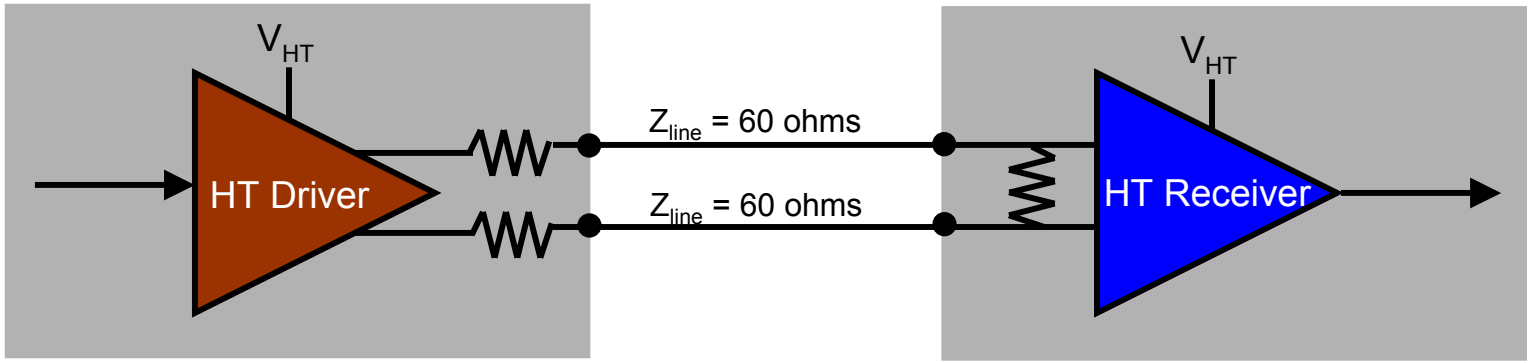
HT Uses Differential Signal pairs

- **Because of the very high clock rates, HyperTransport uses:**
 - Differential signal pairs for CAD, Control, and Clocks.
 - Clocks are source-synchronous, meaning the device driving CAD/Control is also driving Clock(s). This allows longer PC board traces, while maintaining set-up and hold times.
 - One clock (differential pair) is required for each byte of CAD width.

Copyright MindShare Inc., 2002

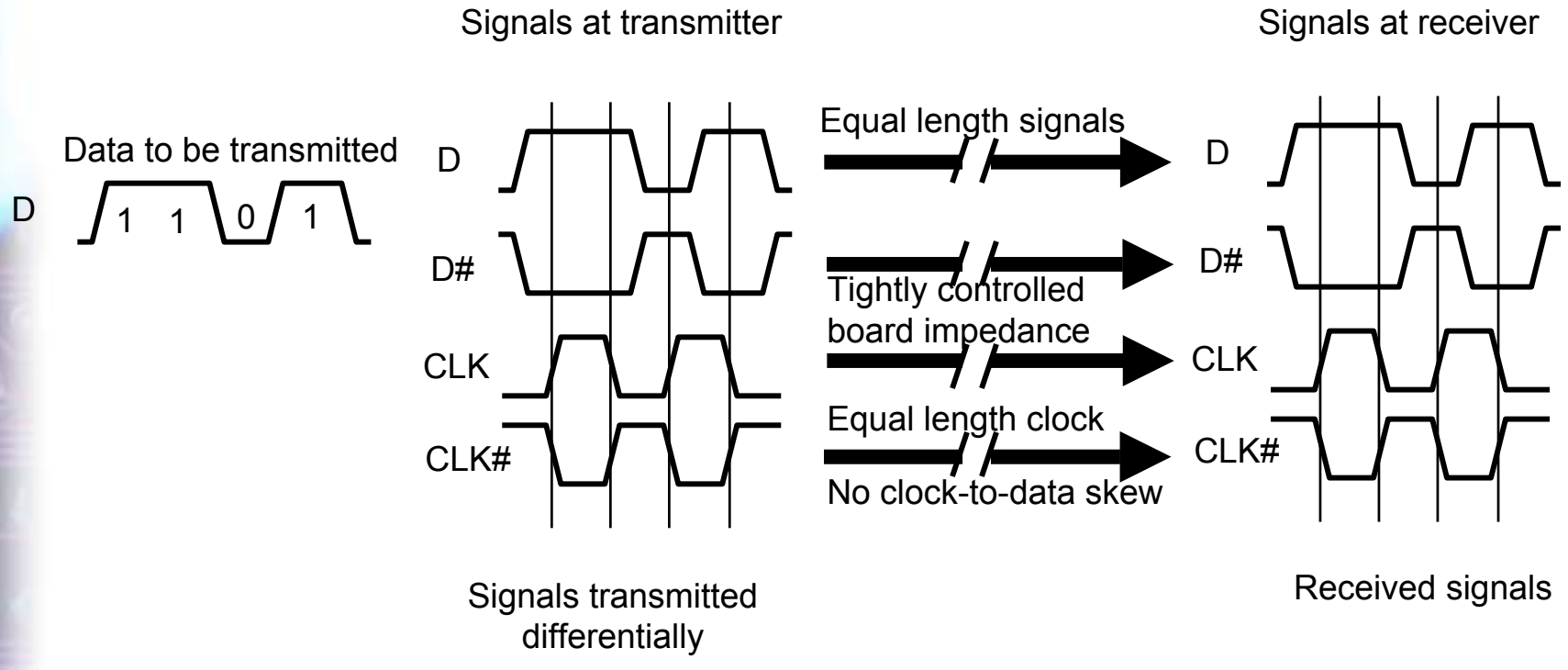
Differential Drivers/Receivers

- Enhanced Low Voltage Differential signaling
 - Characteristic line impedance (Z_{line}) = 60 ohms
 - Driver supply voltage V_{HT}= 1.2 volts
 - Standard 4-layer PCB with controlled impedance sufficient to implement HyperTransport link
 - Trace length of up to 24 inches



Copyright MindShare Inc., 2002

Source Synchronous Clocking



Copyright MindShare Inc., 2002

PlatformConference

Direction • Design • Perspective • Analysis

Scalability:
Link Width And Clock Speed



January 23-24, 2002

Scaling Helps Implementers

- In one protocol, HyperTransport combines a key advantage of both a serial and parallel bus architecture:
 - An implementation that doesn't have high bandwidth requirements can fall back to a low pin count, low clock speed, low power, quasi-serial interface.
 - An implementation that needs all possible bandwidth (e.g. mass-storage devices), can use 32 bit links, running at the higher clock speeds.

Copyright MindShare Inc., 2002

Scalability: Width vs. Pin Count

Link Width →	2	4	8	16	32
Pin Names	**Number of Pins**				
Data Pins (CAD)	8	16	32	64	128
Clock Pins (CLK)	4	4	4	8	16
Control Pins (CTL)	4	4	4	4	4
Subtotal (High Speed)	***16***	***24***	***40***	***76***	***148***
V_{HT}	2	2	3	6	10
GND	4	6	10	19	37
PWROK	1	1	1	1	1
RESET#	1	1	1	1	1
LDTSTOP# / LDTREQ#	2	2	2	2	2
Total Pins / Link Interface	**26**	**36**	**57**	**105**	**199**

Copyright MindShare Inc., 2002

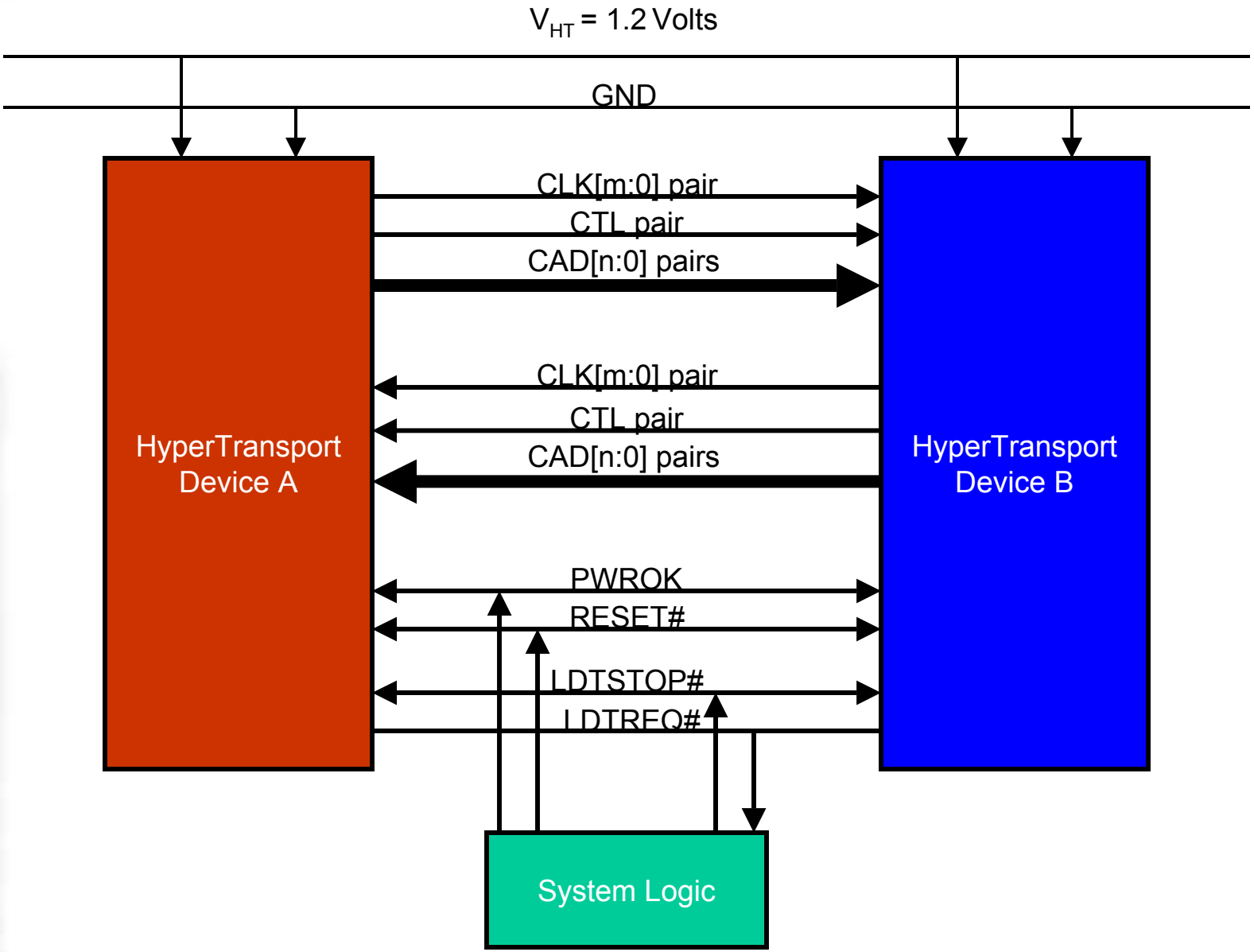
Scalability: Clock Speed

- In the current specification, HyperTransport links allow a maximum clock speed of 800MHz (data is latched on both edges). This means the data rate (per signal) is as high as 1.6 Gbit/s (.2GBytes/s)
- Link bandwidths at various widths and clock speeds:

Link Width (bits)	Bandwidth Per Link (In Gbytes/s) 800 Mhz	400Mhz	200MHz
2	0.8	0.4	0.2
4	1.6	0.8	0.4
8	3.2	1.6	0.8
16	6.4	3.2	1.6
32	12.8	6.4	3.2

Copyright MindShare Inc., 2002

HyperTransport Interface Example



Copyright MindShare Inc., 2002

PlatformConference

Direction • Design • Perspective • Analysis

Protocol Basics: Packet Structure



January 23-24, 2002

HyperTransport Is Packet-Based

- Bus protocol is packet based
 - Packets are multiples of 4-Bytes
 - Packets are transmitted over CAD signals, and contain command, address, data, and other information.
 - Packet types include:
 - Control Packets. Three types:
 - Information packets
 - Request packets
 - Response packets
 - Data packets. Maximum payload is 64 bytes.

Copyright MindShare Inc., 2002

Control Packet Structure

- Information packet (4 byte). Variants:
 - *NOP, Sync/Error*
- Request packet (4 or 8 byte). Variants:
 - *Sized Write (Posted), Broadcast Message, Fence, Sized Write (non-posted), Sized Read, Flush, Fence, Atomic Read-Modify-Write*
- Response packet (4 byte). Variants:
 - *Read Response*
 - *Target Done*

Copyright MindShare Inc., 2002

Data Packet Structure

- Data packet structure varies with the command which caused it:
 - *Sized Dword Read* or *Write* data packets are 1-16 dwords (4-64 bytes)
 - *Sized Byte Read* data packets are 1 dword (any byte combination valid)
 - Sized Byte Write data packets are 1-32 bytes (any byte combination valid)

Copyright MindShare Inc., 2002

HyperTransport Command Codes

Code	Command	Transaction Packet(s)
000000	NOP	Information
000010	Flush	Request
x01xxx	Sized Write	Request/Add/Data
01xxxx	Sized Read	Request/Add
110000	Read Response	Response/Data
110011	Target Done	Response
111010	Broadcast Msg	Request/Address
111100	Fence	Request
111101	Atomic RMW	Request/Add/Data
111111	Sync/Error	Information

Copyright MindShare Inc., 2002

PlatformConference

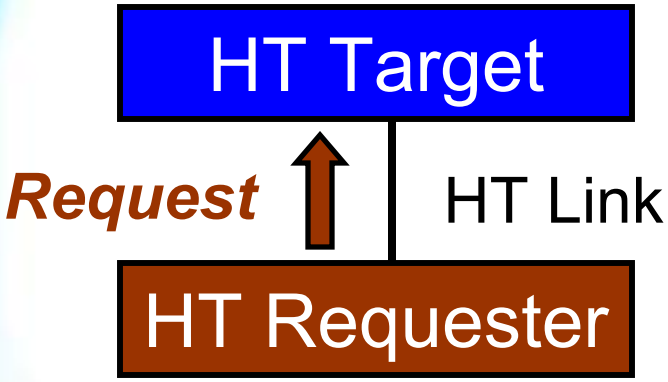
Direction • Design • Perspective • Analysis

Protocol Basics: Requests



January 23-24, 2002

Purpose Of HT Requests

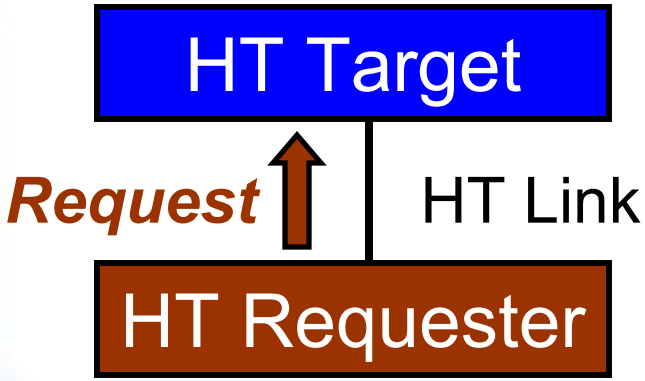


In HT, the sender uses ***Requests*** to indicate:

- It wants to send or forward data (write)
- It wants a target to return data to it (read)
- It wants to begin an *atomic* read/modify/write operation
- It wants additional control over ordering of it's posted transactions (using Flush and Fence commands).
- It wants to broadcast a message to all downstream agents (done by bridges only)

Copyright MindShare Inc., 2002

Requests: Basic Content



While format varies with request type, the basic content of an HT request informs all that see it:

- The target address in system memory or I/O maps
- The request type (command)
- The sender's UnitID
- The amount of data to be transferred (if any)
- Other attributes: byte count, virtual channel, etc.

Copyright MindShare Inc., 2002

Request Classes

- Three classes of HyperTransPort requests
 - Requests that behave like reads, and require a target response and data:
 - Sized (byte and dword) Read, Atomic RMW
 - Requests that behave like writes, and require a target response for confirmation
 - Non-posted Sized (byte and dword) Writes, Flush
 - Requests that behave like writes, and don't require a target response for confirmation
 - Posted Sized (byte and dword) Writes, Broadcast Message, Fence

Copyright MindShare Inc., 2002

Sized (Byte) Read Requests

- Sized (byte) Read transactions transfer any byte combination in a single dword.
- The request packet ***Mask*** field indicates valid bytes in target dword

Copyright MindShare Inc., 2002

Sized (Dword) Read Requests

- Sized (dword) Read transactions are used to read any combination of contiguous dwords in a 16 dword address-aligned group.
- The ***Count*** field in the request indicates total transfer of 1-16 dwords (4-64 bytes).
- All dwords in the range indicated by the starting address plus the value in the ***Count*** field are considered valid.
- Sized Read (dword) command option bits (response passing posted writes, etc) are same as byte reads.

Copyright MindShare Inc., 2002

Sized (Byte) Write Requests

- Sized (byte) Write transactions are used to write any combination of bytes within a naturally aligned 32 byte address range. Start address must be dword aligned.
- The request ***Count*** field indicates the total data transfer size (including the dword write mask).
- The first dword in the DATA packet is a 32-bit write mask which acts as “byte enables”

Copyright MindShare Inc., 2002

Sized (Dword) Write Requests

- Sized (dword) Write transactions are used to write any combination of contiguous dwords in a 16 dword address-aligned group.
- The ***Count*** field in the request indicates total transfer of 1-16 dwords (4-64 bytes).
- All dwords in the range indicated by the starting address plus the value in the ***Count*** field are considered valid (included in the transfer)
- Sized Write (dword) command option bits (write post, etc) are same as byte writes.

Copyright MindShare Inc., 2002

Flush Requests

- Flush is an upstream, non-posted “dummy” read transaction which pushes all posted requests ahead of it to memory before completing.
- If a posted transaction ahead of it is peer-to-peer, a FLUSH only guarantees it has been pushed to the destination host bridge.
- A response is required indicating completion of the Flush operation, no data is returned.

Copyright MindShare Inc., 2002

Fence Requests

- Fence is used to help manage traffic:
 - Fence is a barrier between posted writes; it affects all channels and IO streams. It is issued from a device to a bridge, or from bridge to bridge.
 - It uses the posted request channel, so no response is expected or returned. If it's PassPW bit = 0, it won't pass any posted channel requests from any UnitID.
 - Other packets with PassPW = 1 may pass a Fence; otherwise they won't.

Copyright MindShare Inc., 2002

Atomic RMW Requests

- Atomic Read-Modify-Write (ARMW):
 - Atomic RMW is used so that a memory location may be read, modified, and conditionally written back—all without the race-condition of another device trying to doing it at the same time.
 - The HT Atomic RMW is a hybrid: it starts as a non-posted request, but has a read response. The target address remains locked to other devices until it fully completes.
 - HyperTransport supports two ARMW variants:
 - Fetch and Add
 - Compare and Swap

Copyright MindShare Inc., 2002

Broadcast Message Requests

- ***Broadcast Message*** requests have the following features:
 - They are initiated by the host bridge, and used to send messages to all agents.
 - The request travels downstream in the posted-write channel the entire length of the chain, and is accepted and forwarded by all devices.
 - The 8 byte address field of the request uses ranges reserved and recognized by all devices
 - The message content is also carried in the address field of the Broadcast Message request. There is no data phase.

Copyright MindShare Inc., 2002

PlatformConference

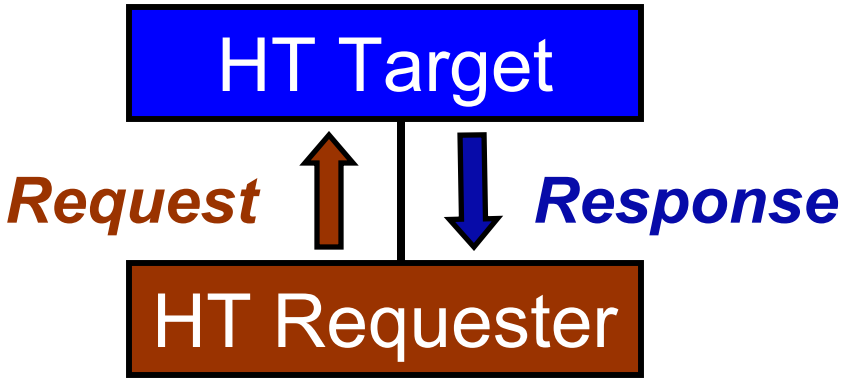
Direction • Design • Perspective • Analysis

Protocol Basics: Responses/Data



January 23-24, 2002

The Purpose Of HT Responses

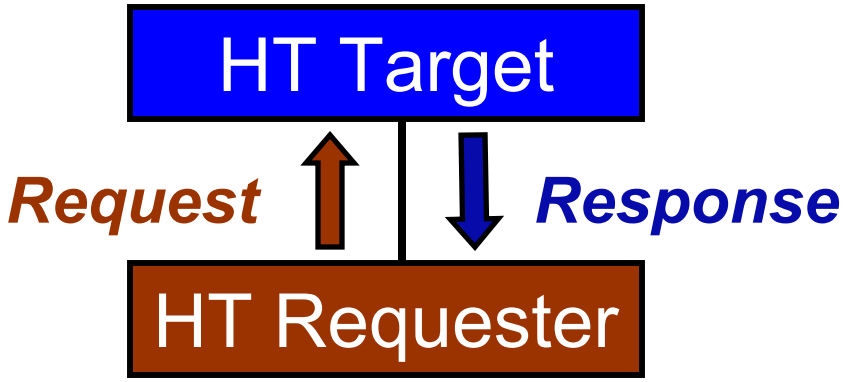


In HyperTransport, if a ***Response*** follows a request; the target indicates it wants to:

- Return data to satisfy an earlier read or Atomic read-modify-write (RMW) request
- Confirm the arrival of non-posted write data.
- Confirm the completion of a Flush operation
- Report errors

Copyright MindShare Inc., 2002

HT Responses: Basic Content



While format varies with request type, basic content of an HT response informs all devices in it's path:

- The response type (command)
- The response direction (upstream or downstream)
- UnitID & Source Tag information
- The amount of data to be transferred (if any)
- Misc information: error, virtual channel,etc

Copyright MindShare Inc., 2002

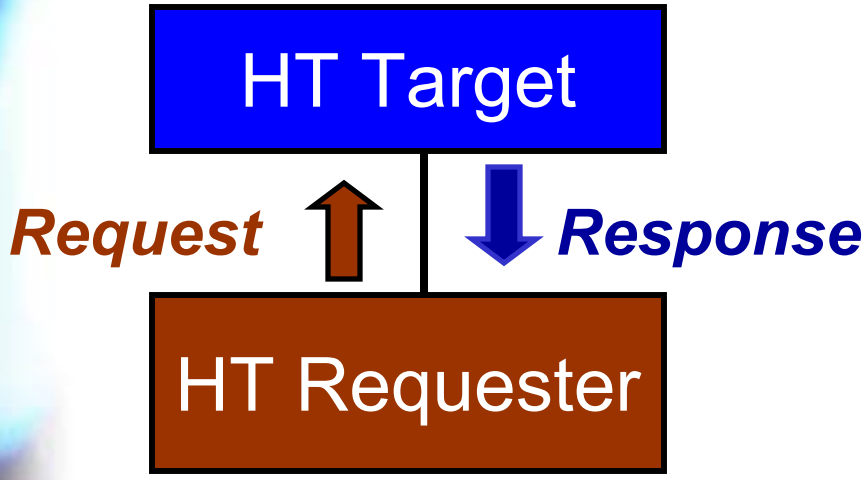
Read Request & Response/Data



- The Requester issues a Sized Read or Atomic RMW request packet.
- The HT target drives the four byte response when it has status and all requested data available

Copyright MindShare Inc., 2002

Non-Posted Write & Response

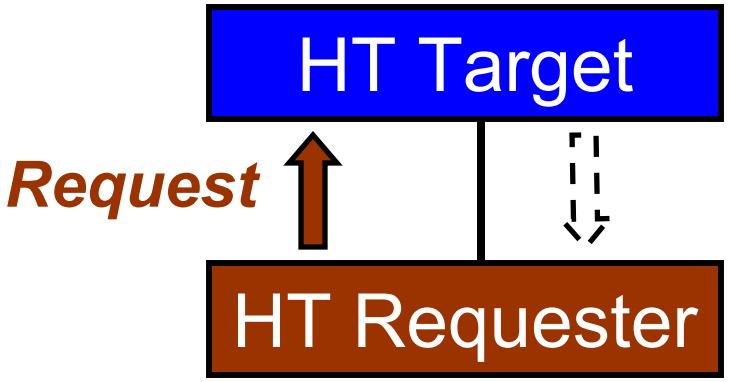




- The Requester issues a non-posted Sized Write (byte or dword) request.
- The HT target drives the four byte TgtDone response when it has status available

Copyright MindShare Inc., 2002

Posted Write: No Response

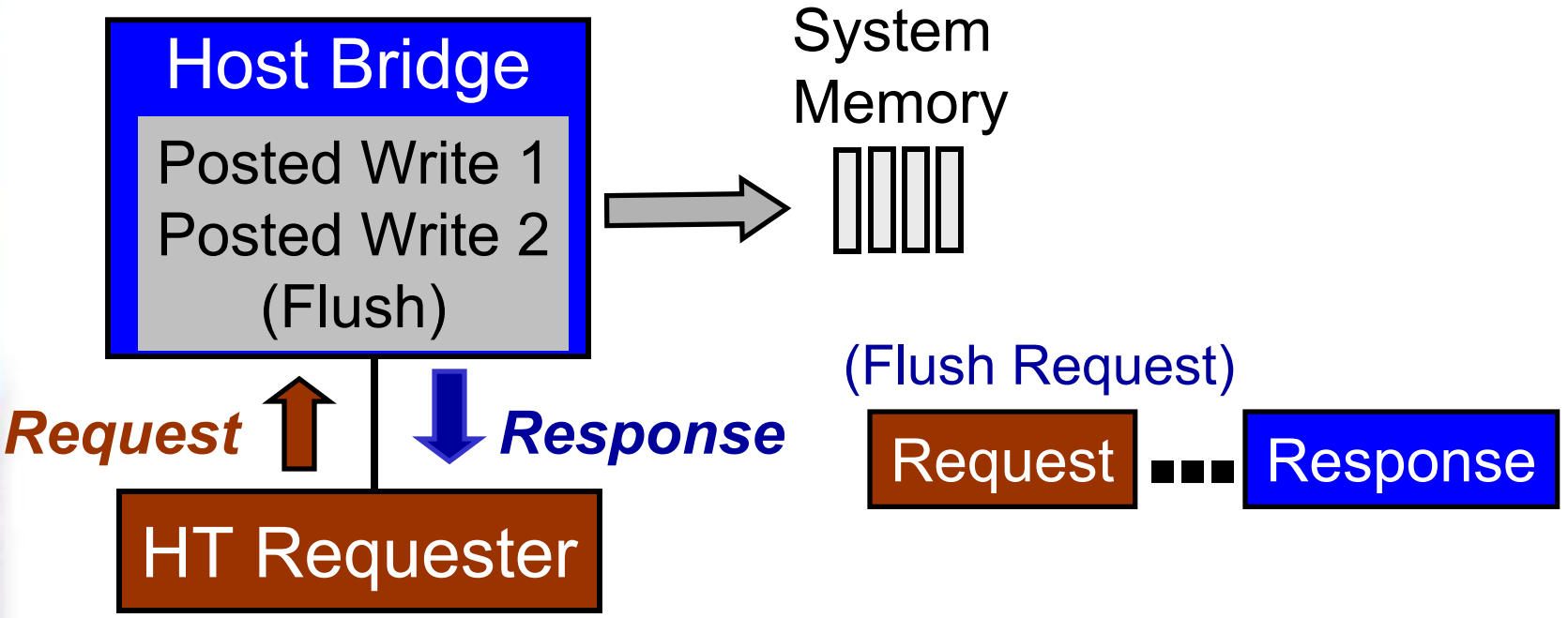




- The Requester issues a posted Sized Write request, followed by data.
- While faster (no response), we don't know exactly when transaction completes or if there was an error.

Copyright MindShare Inc., 2002

Flush Request Has A Response

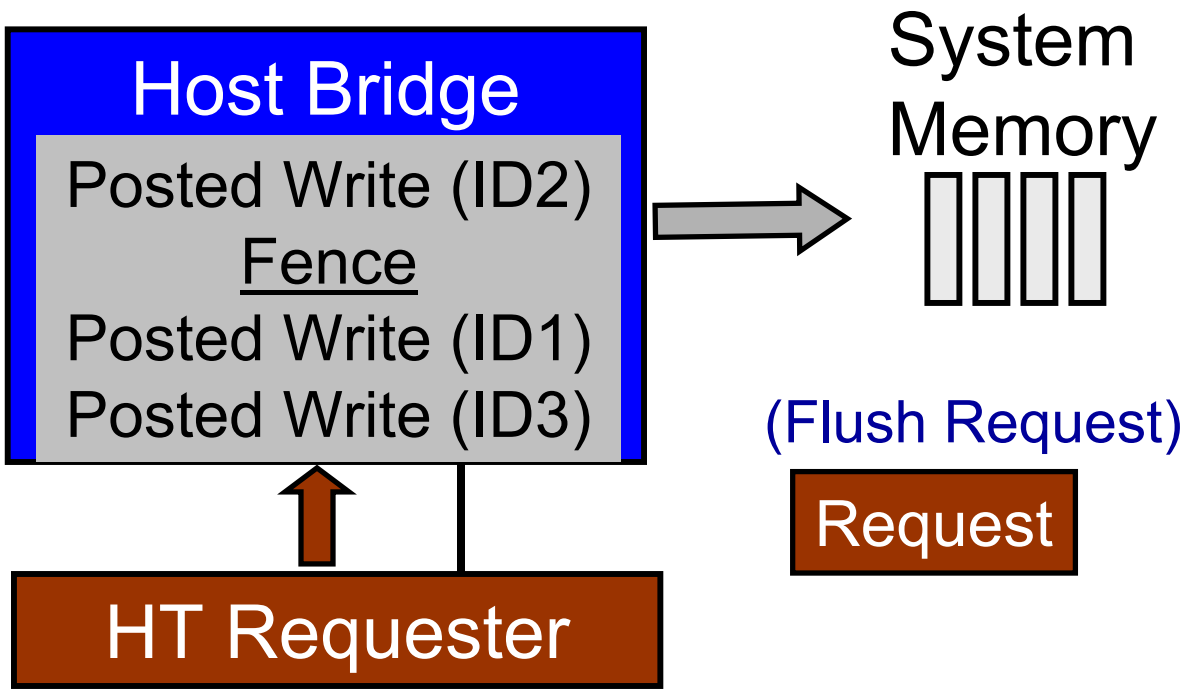


- The Requester issues a Flush request packet.
- The HT target (Host Bridge) drives the four byte TgtDone response when previously posted writes have reached memory.

Copyright MindShare Inc., 2002

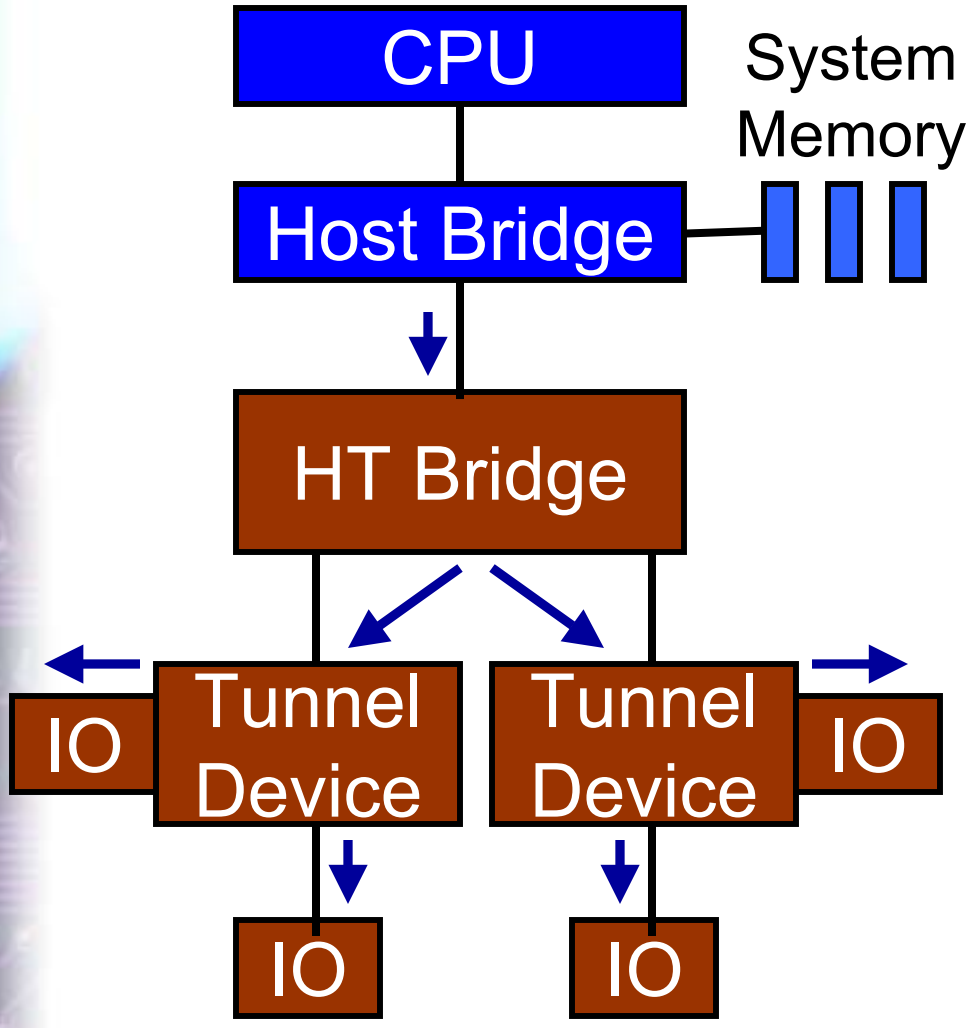
Fence Request

- Without Fence, posted writes from different UnitID's are handled as separate streams.
- Fence blocks subsequent packets (from all UnitID's) without PassPW = 1 while it pushes posted writes ahead of it to memory.



Copyright MindShare Inc., 2002

Broadcast Message



Example:

- Host issues Broadcast Message request
- HT-HT bridge reissues this request on all of it's downstream links
- Tunnel devices always accept and pass these requests same direction
- broadcast message has been routed and accepted on all links.

Copyright MindShare Inc., 2002

PlatformConference

Direction • Design • Perspective • Analysis

Managing The Links: PCI Configuration



January 23-24, 2002

Managing Links: PCI Configuration

- At boot time, PCI configuration is used to initially set-up HyperTransport devices:
 - Read in configuration information about device requirements and capabilities.
 - Program the device with address range, error handling policy, etc.
- Bridges, tunnels, and single-link peripherals have different responsibilities (in forwarding transactions, etc), so have different configuration capability block formats.

Copyright MindShare Inc., 2002

PCI Configuration, Continued

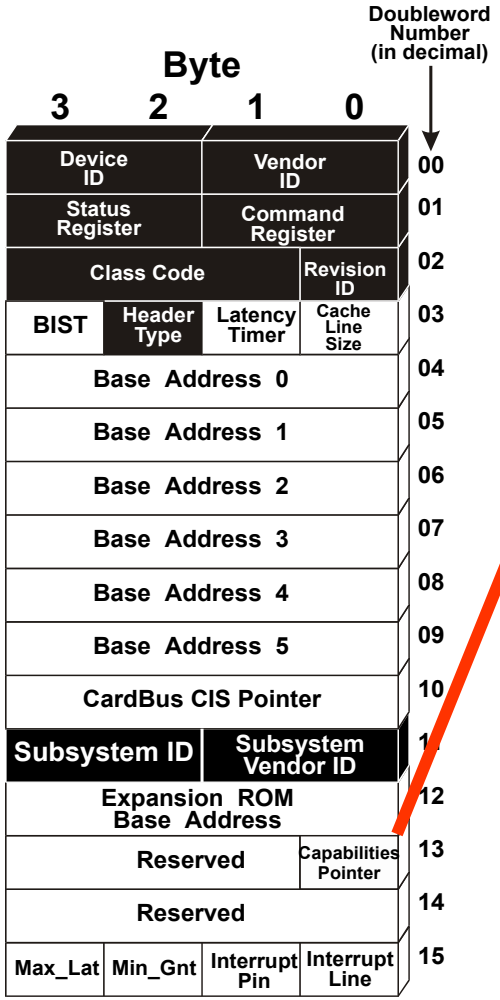
- HyperTransport PCI Configuration:
 - Uses the 256 byte PCI Configuration Space
 - Uses different header formats for bridges
 - Has it's own Capability Registers set
 - Uses Configuration Transactions, type 0-1
 - Uses same bus numbering scheme as PCI
 - Maps all Configuration Space into the 40-bit HyperTransport memory map.

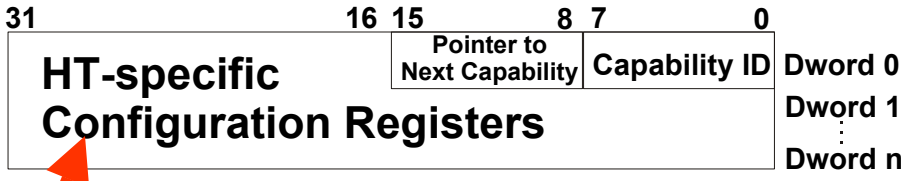
Base Address	Top Address	Size	Usage
FD FE00 0000	FD FFFF FFFF	32 MB	Configuration

Copyright MindShare Inc., 2002

HT PCI Capability Register Set

- HT specific registers are mapped in a device's PCI Configuration Space using the capability list pointer.





- Block format varies:
 - Single-link peripheral has a single primary block with one Link Control register
 - A tunnel has a single primary block with two Link Control registers

Copyright MindShare Inc., 2002

PlatformConference

Direction • Design • Perspective • Analysis

Managing The Links: Flow Control



January 23-24, 2002

Managing Links: Flow Control

- HyperTransport requires the transmitter on each link to accept NOP packets from it's receiver at reset indicating buffer capacity, then establish a packet coupon scheme that:
 - Guarantees no transmitter will send a packet that the receiver can't accept
 - Eliminates the need for inefficient disconnects and retries on the link.
 - Requires each receiver to dynamically inform the transmitter (via NOP packets) as buffer space becomes available.

Copyright MindShare Inc., 2002

Virtual Channels And Flow Control

- HyperTransport supports three separate virtual channels, meaning that 6 separate flow control buffers (per link) are needed:
 - Non-posted request buffer
 - Posted request buffer
 - Response buffer
 - Non-posted data buffer
 - Posted data buffer
 - Response data buffer
- Flow control requires that receiver (NOP) information keep the transmitter apprised of the dynamic state of all of these.

Copyright MindShare Inc., 2002

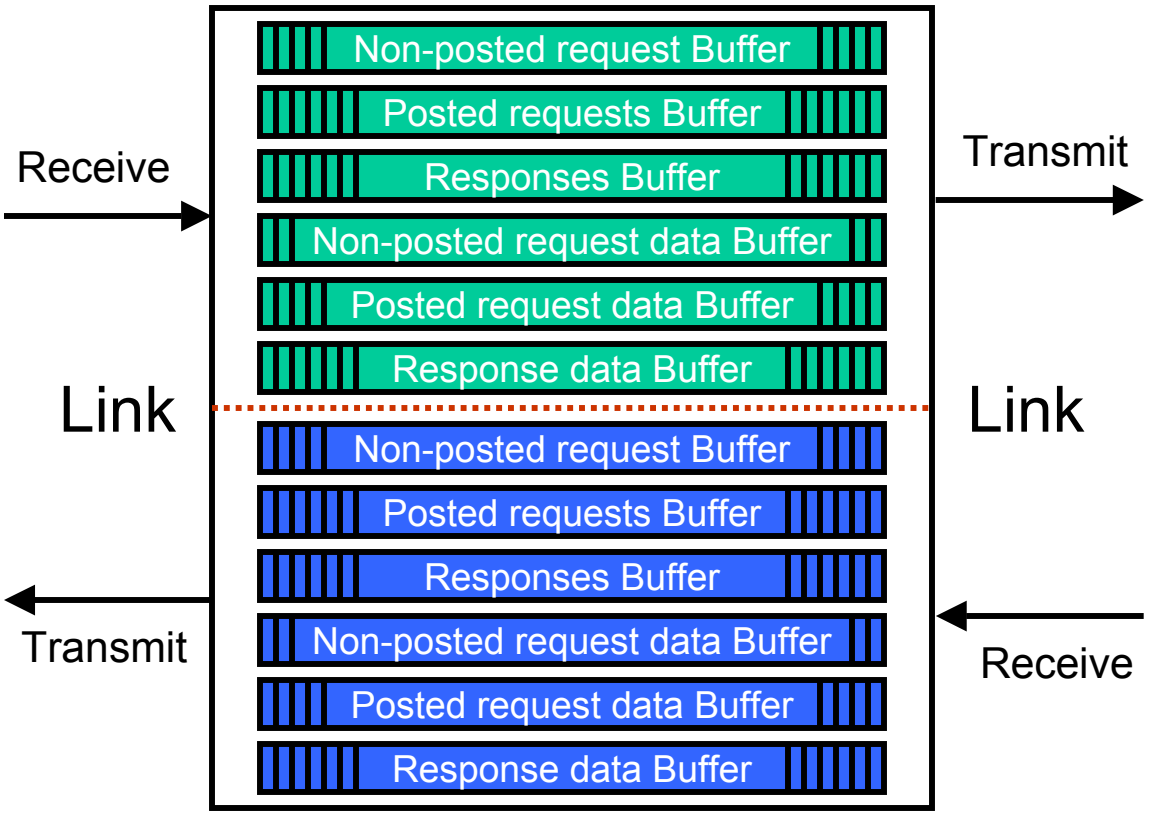
Isochronous Transactions

- HyperTransPort provides for an optional, highest prioroty, isochronous (ISOC) virtual channel to handle for time-critical (e.g. real-time) data.
- ISOC devices implement dedicated flow control buffers to support it.
- ISOC bit set in request and response packets indicates ISOC virtual channel routing.
- If the ISOC bit is set, all link devices which support it will use this channel; others will pass it along in posted or non-posted channels.

Copyright MindShare Inc., 2002

Receiver Flow Control Buffers

- This diagram illustrates the receiver's use of 6 buffers to support flow control (device doesn't support isochronous channel).



Copyright MindShare Inc., 2002

Identifying IO Streams

- The UnitID field within a request or response packet is used to identify node IO streams.
- Which node UnitID used is summarized below:

	Upstream	Downstream
Request Packet	UnitID of source node	UnitID of host bridge:0
Response Packet	UnitID of target node	UnitID of source node

- Host bridge is responsible for managing interactions between streams and maintaining ordering within streams

Copyright MindShare Inc., 2002

Virtual Channels

- 3 Virtual Channels
 - Posted Requests
 - Non-Posted Requests
 - Responses
- Separate command / data buffer implemented within devices for each virtual channel
- Ordering rules prevent transactions in different virtual channels (same IO stream) from blocking each other.

Copyright MindShare Inc., 2002

PlatformConference

Direction • Design • Perspective • Analysis

Managing The Links: Error Handling



January 23-24, 2002

Error Handling Options

- HyperTransport defines required and optional error checking:
 - Cycle Redundancy Check (CRC) generation and checking on links is required.
 - Error logging in HyperTransport device PCI Configuration Space is defined, but optional. If not implemented, these bits are hardwired to 0.

Copyright MindShare Inc., 2002

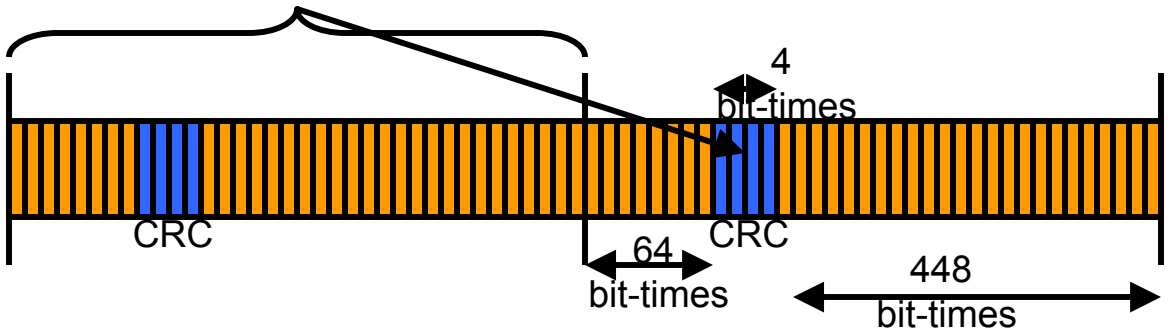
Purpose Of HyperTransport CRC

- Features of CRC:
 - Catches link transmission errors
 - 32-bit CRC is independently calculated and checked on each 8 bits of CAD.
 - CTL is included in CRC check for lowest 8 bits of CAD
 - CRC calculated over 512 bit times on link, without regard to packets, and sent during a prescribed offset in CRC “window”.

Copyright MindShare Inc., 2002

Transmission Of CRC

- CRC error checking mandatory
- CRC computed over 512 bit-time window
- 32-bit CRC value is sent at prescribed offset in CRC “window”



Copyright MindShare Inc., 2002

In the Event Of A CRC Error

- Depending on the nature of the CRC error detected, the CRC detector in the receiver may invoke:
 - Fatal or non-fatal interrupts
 - Sync Flooding (catastrophic and having a similar effect as PCI SERR#)

Copyright MindShare Inc., 2002

Other HyperTransport Errors

- Protocol (violation) errors
- Receive Buffer Overflow errors
- End Of Chain errors
- Chain Down error
- Response Error

Copyright MindShare Inc., 2002

How These Errors Are handled

- Depending on the type of error, devices may take several actions. Listed in order of increasing severity:
 - Send a response with error bit(s) set
 - Generate an error interrupt (could be in addition to sending an error response)
 - Use Sync Flooding. This is used when no other method of reporting is reliable, and brings the entire chain to inactive state (a reset is required).
 - If Configuration Space Error Handling CSR is in use, errors are also placed there so software may determine nature of the problem

Copyright MindShare Inc., 2002

Platform Conference

Direction • Design • Perspective • Analysis

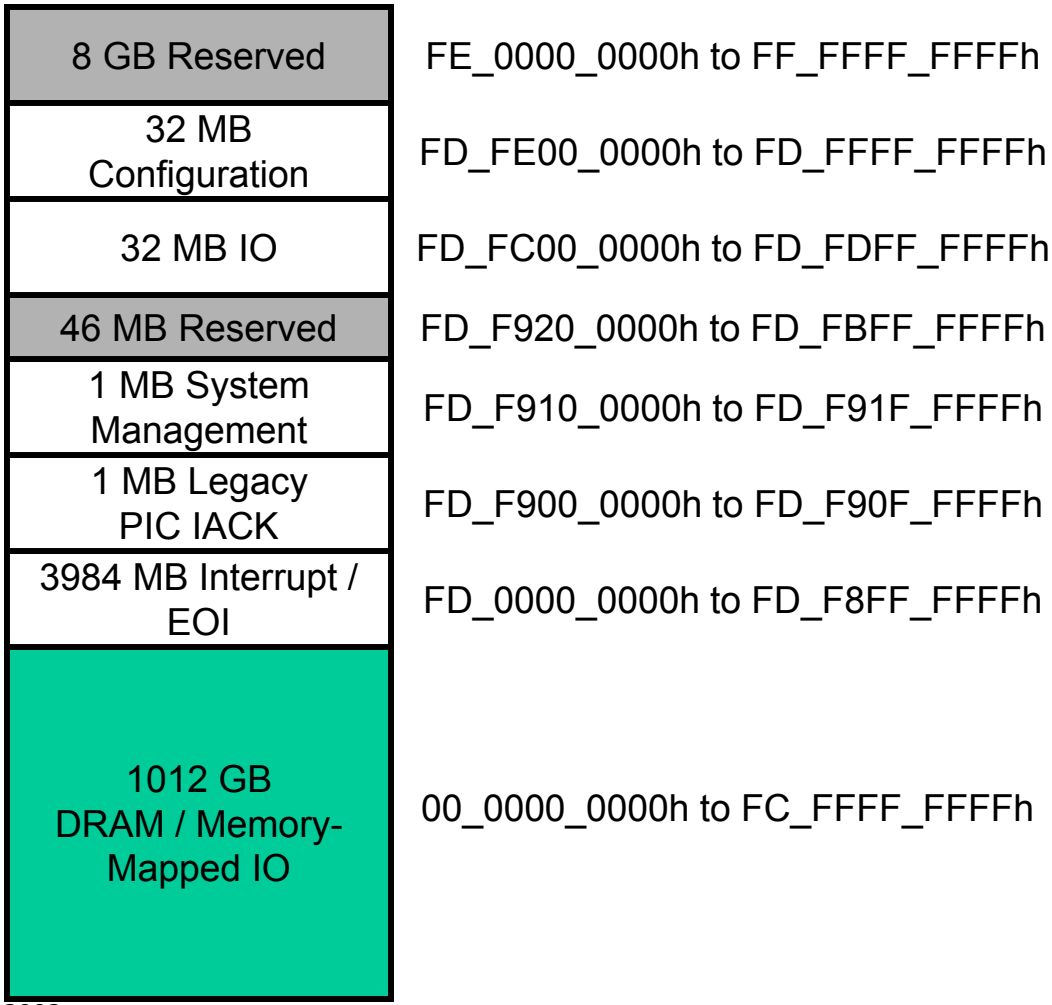
HyperTransport Address Map



January 23-24, 2002

Address Map

- 40-bit (1Terabyte) memory address space



Copyright MindShare Inc., 2002